

# 11\_3\_1

# UPGRADE

일시 2023년 10월 11일

학번 2021763022

이름 김정원

5초 동안 공개되는 문자들의 조합을 기억해 맞추는 게임

# 문자 퍼즐

문자 퍼즐 맞추기

격자모양의 판에 숨겨진 두 개의 문자를 찾는  
게임입니다.

행의 숫자(4 또는 6)를 입력하고 Enter>



**Dynamic**  
플레이어 추가

## Introduction 2 PLAYER

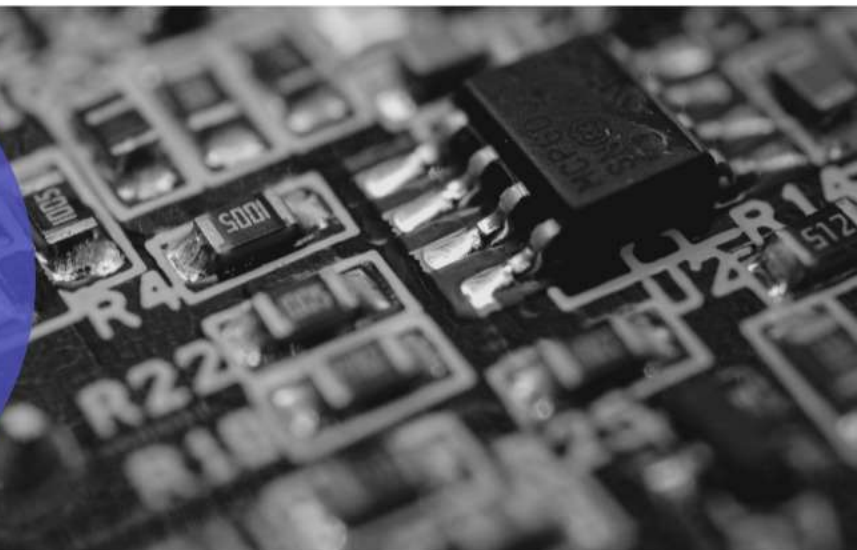
- 2명의 플레이어로 게임다운 요소를 더함
- 플레이어에 따른 턴, 점수 표시
- 게임 종료 시 승패와 플레이 시간 등 표시

## Repair & Error correction

- 게임 시작 화면 추가
- 색상, 그리드를 수정하여 가시성을 높임
- 에러 요소 방지



**Repair**  
코드 보수



# Start Game!

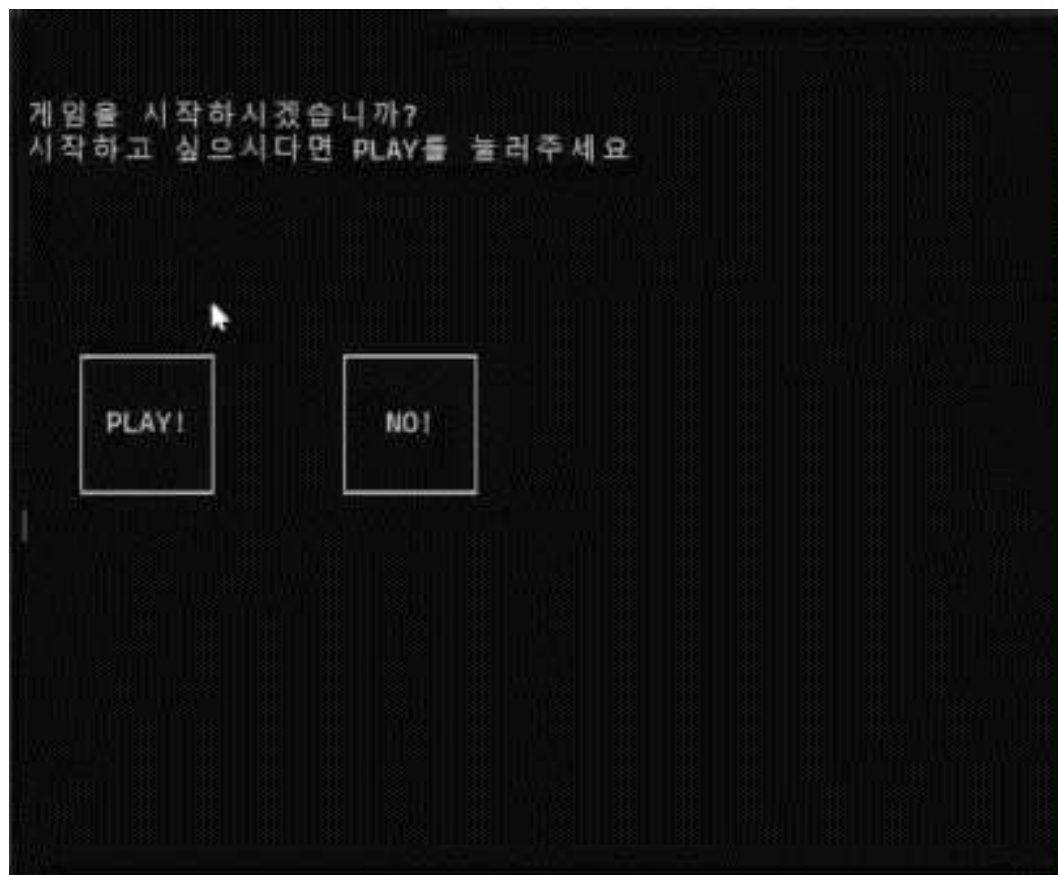
1. 플레이 버튼 구현
2. 게임 규모 설정

```
int main(void){
    int n, countA =0;
    int countB=0;
    int turn=1;
    char r;

    char question[10][10]={0};
    char answer[10][10]={0};
    long pst, seconds, minutes;
    clock_t start, end;
    srand(time(NULL));

    r= AU_ready();
    system("cls");

    if(r!='Y'){
        printf("\n\n\n\n\n게임을 종료합니다. \n\n");
        return 0;
    }else{
        n=intro_game();
        system("cls");
    }
}
```







1. 플레이 버튼 구현
2. 게임 규모 설정

```
int main(void){
    int n, countA =0;
    int countB=0;
    int turn=1;
    char r;

    char question[10][10]={0};
    char answer[10][10]={0};
    long pst, seconds, minutes;
    clock_t start, end;
    srand(time(NULL));

    r= AU_ready();
    system("cls");

    if(r!='Y'){
        printf("\n\n\n\n\n게임을 종료합니다. \n\n");
        return 0;
    }else{

        n=intro_game();
        system("cls");
```

```
int intro_game(void)
{
    int n;
    system("cls");
    printf("문자 퍼즐 맞추기 \n\n");
    printf("격자 모양의 판에 숨겨진 두 개의 문자를 찾는 \n");
    printf("게임입니다. \n\n");
    printf("행의 숫자 (4 또는 6)를 입력하고 Enter>");
    scanf("%d",&n);

    if(n==4 || n==6){
        return n;
    }else{
        system("cls");
        printf("문자 퍼즐 맞추기 \n\n");
        printf("격자 모양의 판에 숨겨진 두 개의 문자를 찾는 \n");
        printf("게임입니다. \n\n");
        printf("행의 숫자 (4 또는 6)를 입력하고 Enter>");
        printf("4 또는 6을 입력해주세요.");
        Sleep(1500);

        return intro_game();
    }
}
```

r게임 규모를 설정하기 위해 4또는 6의 숫자를 받습니다.  
이 외의 숫자나 문자를 받을 경우 경고 후, 다시 숫자를 받습니다.

## Before\_1Player

0	1	2	3
B	5	6	7
8	9	10	B
12	13	14	15

맞은 개수: 1개(총 8개)

첫 번째 숫자를 입력하고 Enter>  
두 번째 숫자를 입력하고 Enter>11  
맞았습니다. 아무키나 누르면 다시 시작합니다.

## After\_2Player

0	1	B	3
H	5	A	F
8	9	10	H
B	F	A	15

[플레이어 B]  
총 문제 개수: 8

맞은 개수: 1개

맞은 개수: 3개

첫 번째 숫자를 입력하고 Enter>6  
두 번째 숫자를 입력하고 Enter>14  
맞았습니다. 엔터를 누르면 넘어갑니다.

```

void game_control(char question[][10], int n, int *countA, int *countB, int *turn)
{
    int user1, user2, row1, row2, col1, col2;
    int T = *turn;

    if(T%2==0){
        gotoxy(28, 2);
        printf("[플레이어 B]");
    }else{
        gotoxy(28, 2);
        printf("[플레이어 A]");
    }

    gotoxy(2, 14);
    printf("첫 번째 숫자를 입력하고 Enter>");
    scanf("%d", &user1);
    row1=user1/n;
    col1=user1%n;

    gotoxy(2, 15);
    printf("두 번째 숫자를 입력하고 Enter>");
    scanf("%d", &user2);
    row2=user2/n;
    col2=user2%n;

    if (question[row1][col1]==question[row2][col2]){
        if(T%2==1){
            *countA=*countA+1;
            gotoxy(28, 2);
        }else{
            *countB=*countB+1;
            gotoxy(28, 2);
        }

        gotoxy(28, 5);
        printf(ANSI_COLOR_WHITE "맞은 개수 :%2d개 ", *countA);
        gotoxy(28, 7);
        printf(ANSI_COLOR_WHITE "맞은 개수 :%2d개 ", *countB);

        if(T%2==0){
            gotoxy(3+4*col1, 2+2*row1);
            printf(ANSI_COLOR_CYAN "%2c", question[row1][col1]);
            gotoxy(3+4*col2, 2+2*row2);
            printf(ANSI_COLOR_CYAN "%2c", question[row2][col2]);
        }else{
            gotoxy(3+4*col1, 2+2*row1);
            printf(ANSI_COLOR_RED "%2c", question[row1][col1]);
            gotoxy(3+4*col2, 2+2*row2);
            printf(ANSI_COLOR_RED "%2c", question[row2][col2]);
        }
    }
}

```

## After\_2Player

0	1	B	3
H	5	A	F
8	9	10	H
B	F	A	15

[플레이어 B]  
총 문제 개수: 8

맞은 개수: 1개

맞은 개수: 3개

첫 번째 숫자를 입력하고 Enter>6  
두 번째 숫자를 입력하고 Enter>14  
맞았습니다. 엔터를 누르면 넘어갑니다.



# Correct

0	1	2	3
A	5	6	7
8	9	10	11
12	A	14	15

[플레이어 B]  
총 문제 개수: 8  
맞은 개수: 1개  
맞은 개수: 0개

첫번째 숫자를 입력하고 Enter>

# Wrong

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

[플레이어 A]  
총 문제 개수: 8  
맞은 개수: 0  
맞은 개수: 0

첫번째 숫자를 입력하고 Enter>

# Game Over...

1. 승패 여부 표시
2. 게임 종료시간 표시

```
do{
    game_control(question, n, &countA, &countB, &turn);
}while(countA+countB<n*n/2);

gotoxy(2, 18);
printf("모 두 맞 았 습 니 다 . 종 료 합 니 다 . \n");
end=clock();
pst=(end-start)/CLK_TCK;
minutes=pst/60;
seconds=pst-minutes*60;
gotoxy(2, 17);
if(countA>countB){
    printf("플 레 이 어 A 승 리 !!");
}else if(countA<countB){
    printf("플 레 이 어 B 승 리 !!");
}else{
    printf("무 승 부 ");
}

gotoxy(2, 19);
printf("경 과 시 간 : %ld분 %ld 초 \n", minutes, seconds);

return 0;
```

G	E	B	C
H	D	A	F
D	C	G	H
B	F	A	E

[플레이어 B]  
총 문제 개수: 8

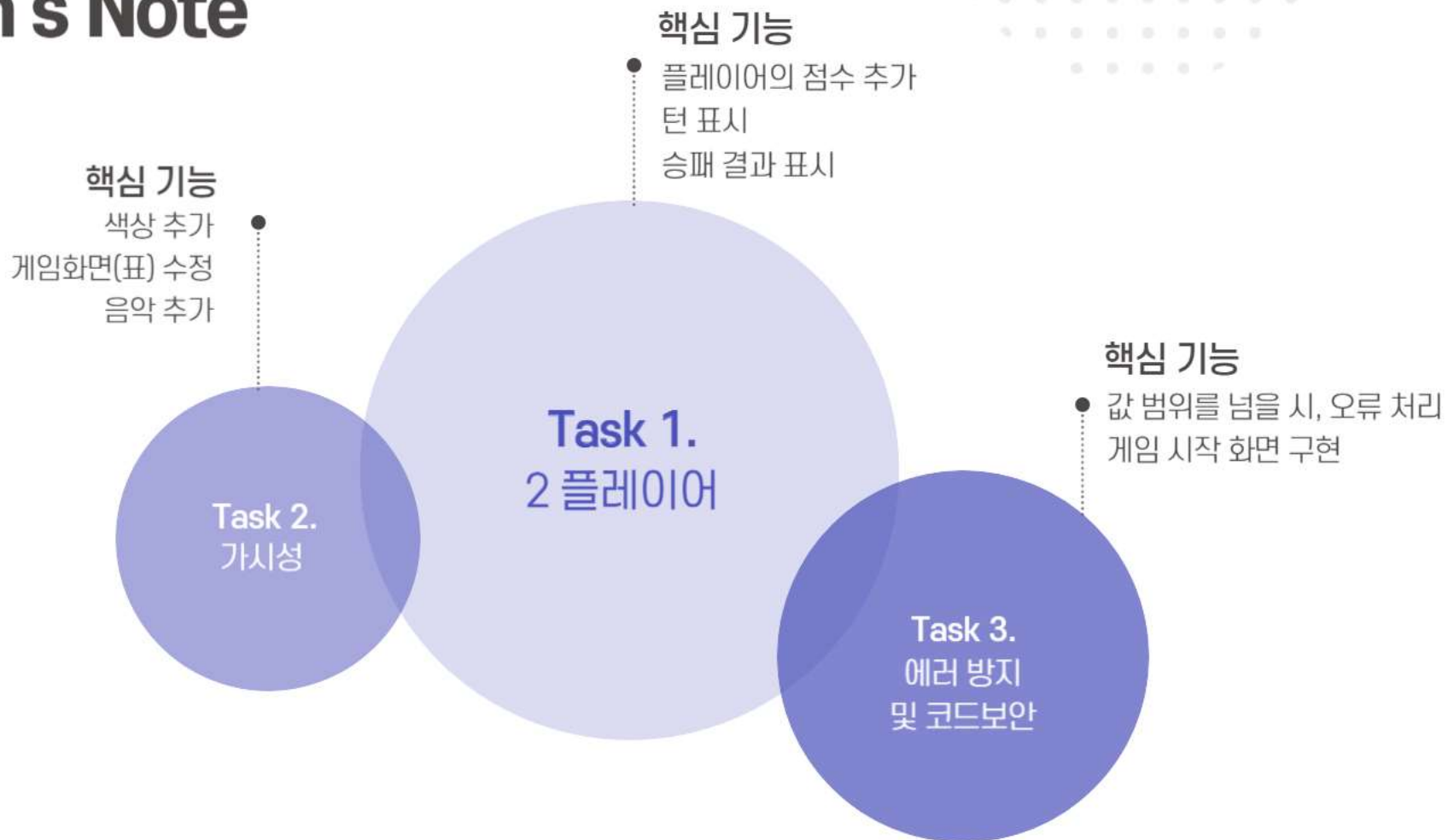
맞은 개수: 3개

맞은 개수: 5개

플레이어 B 승리!!  
모두 맞았습니다. 종료합니다.  
경과시간 : 1분 25 초

-----  
Process exited after 101.6 seconds with return value 0  
계속하려면 아무 키나 누르십시오 . . .

# Garden's Note



게임 프로그래밍

# THANK YOU

김정원 2021763022

2023-10-11

게임 프로그래밍

# THANK YOU

김정원 2021763022

2023-10-11